

(11)Publication number : 10-091564
(43)Date of publication of application : 10.04.1998

G06F 13/10
G06F 13/10 . . .

(72)Inventor : SENATOR STEVEN T
PASSMORE DALE R
GITTINS ROBERT S

Priority number : 96 632006 Priority date : 15.04.1996 Priority country : US

The diagram illustrates a computer system architecture with the following components and connections:

- Top Section (20):** Contains a **DE** (Data Entry) block, an **UPS** (Uninterruptible Power Supply) block, and a **ユーティリティプログラム** (Utility Program) block.
- Section 28:** A shaded rectangular block below the top section.
- Section 30:** Contains three blocks labeled **ユーティリティ** (Utility), **システム** (System), and **ユーティリティ** (Utility).
- Section 32:** A shaded rectangular block below section 30.
- Section 34:** A block labeled **ユーティリティプログラム** (Utility Program).
- Section 36:** A shaded rectangular block below section 34.
- Section 38:** A block labeled **ユーティリティプログラム** (Utility Program).
- Section 40:** A shaded rectangular block below section 38.
- Section 42:** A block labeled **ユーティリティプログラム** (Utility Program).
- Section 44:** A block labeled **ユーティリティプログラム** (Utility Program).
- Section 46:** A block labeled **ユーティリティプログラム** (Utility Program).
- Section 48:** A block labeled **ユーティリティプログラム** (Utility Program).
- Section 50:** A block labeled **ユーティリティプログラム** (Utility Program).
- Section 52:** A block labeled **ユーティリティプログラム** (Utility Program).
- Section 54:** A block labeled **ユーティリティプログラム** (Utility Program).
- Section 56:** A block labeled **ユーティリティプログラム** (Utility Program).
- Section 58:** A block labeled **ユーティリティプログラム** (Utility Program).
- Section 60:** A block labeled **ユーティリティプログラム** (Utility Program).
- Section 62:** A block labeled **ユーティリティプログラム** (Utility Program).
- Section 64:** A block labeled **ユーティリティプログラム** (Utility Program).
- Section 66:** A block labeled **ユーティリティプログラム** (Utility Program).
- Section 68:** A block labeled **ユーティリティプログラム** (Utility Program).
- Section 70:** A block labeled **ユーティリティプログラム** (Utility Program).
- Section 72:** A block labeled **ユーティリティプログラム** (Utility Program).
- Section 74:** A block labeled **ユーティリティプログラム** (Utility Program).
- Section 76:** A block labeled **ユーティリティプログラム** (Utility Program).
- Section 78:** A block labeled **ユーティリティプログラム** (Utility Program).
- Section 80:** A block labeled **ユーティリティプログラム** (Utility Program).
- Section 82:** A block labeled **ユーティリティプログラム** (Utility Program).
- Section 84:** A block labeled **ユーティリティプログラム** (Utility Program).
- Section 86:** A block labeled **ユーティリティプログラム** (Utility Program).
- Section 88:** A block labeled **ユーティリティプログラム** (Utility Program).
- Section 90:** A block labeled **ユーティリティプログラム** (Utility Program).
- Section 92:** A block labeled **ユーティリティプログラム** (Utility Program).
- Section 94:** A block labeled **ユーティリティプログラム** (Utility Program).
- Section 96:** A block labeled **ユーティリティプログラム** (Utility Program).
- Section 98:** A block labeled **ユーティリティプログラム** (Utility Program).
- Section 100:** A block labeled **ユーティリティプログラム** (Utility Program).

[Date of request for examination]
[Date of sending the examiner's decision of rejection]
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]
[Date of final disposal for application]
[Patent number]
[Date of registration]
[Number of appeal against examiner's decision of rejection]
[Date of requesting appeal against examiner's decision of rejection]
[Date of extinction of right]

03/09/22

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-91564

(43) 公開日 平成10年(1998) 4月10日

(51) Int.Cl.⁶

G 0 6 F 13/10

識別記号

3 2 0

3 3 0

F I

G 0 6 F 13/10

3 2 0 A

3 3 0 B

審査請求 未請求 請求項の数 9 O L (全 11 頁)

(21) 出願番号 特願平9-97141

(22) 出願日 平成9年(1997) 4月15日

(31) 優先権主張番号 6 3 2 0 0 6

(32) 優先日 1996年4月15日

(33) 優先権主張国 米国 (U S)

(71) 出願人 597004720

サン・マイクロシステムズ・インコーポレ
ーテッド

Sun Microsystems, In
c.

アメリカ合衆国カリフォルニア州94043-
1100, マウンテン・ビュー, ガーシア・ア
ヴェニュー 2550, エムエス・ピーエイエ
ル1-521

(74) 代理人 弁理士 社本 一夫 (外5名)

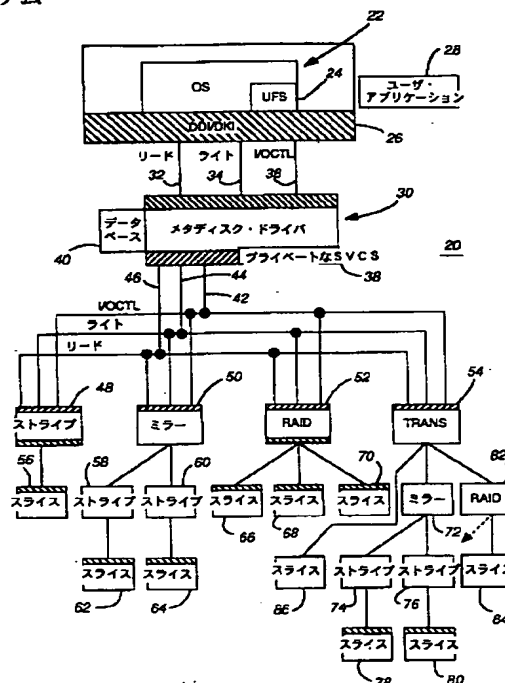
最終頁に続く

(54) 【発明の名称】 複数の独立なデバイス・ドライバを組み入れたコンピュータ・システムのためのメタデバイス・
ドライバをリネーム/交換する方法及びシステム

(57) 【要約】

【課題】 特定のドライバに依存せずにドライバのリネーム及び交換機能を有するメタディスク・ドライバを提供すること。

【解決手段】 本発明のメタディスク・ドライバ (30) は、コンピュータ・オペレーティング・システム (22) と、1つ又は複数のメタドライバ及び下位の階層化されたドライバとの中間にあり、ドライバのリネーム及び交換機能を実行する。この機能は、下位のデバイスがオンラインであり、従って、動作が完了するか又はいずれかの割り込みの生じる間に行われない限り回復可能である原子的動作から構成されている間に、実行することができる。



【特許請求の範囲】

【請求項1】 個別の及び相互に関連する属性を有する複数のドライバを含むコンピュータ・システムにおけるドライバ・ヒエラルキを、前記複数のドライバの他のものに対して変更する方法であって、
前記ドライバの中の特定の1つにその個別の属性を変更するように命令するステップと、
前記ドライバの中の前記特定の1つと前記ドライバの中の前記特定の1つの親類とにサービスを固定するステップと、
前記ドライバの中の前記特定の1つの前記個別の属性と前記ドライバの中の前記特定の1つの前記親類のそれぞれの前記相互に関連する属性とを変更するステップと、
前記ドライバの中の前記特定の1つと前記ドライバの中の前記特定の1つの前記親類とへのサービスの固定を解除するステップと、
を含むことを特徴とする方法。

【請求項2】 個別の及び相互に関連する属性を有する複数のドライバを含むコンピュータ・システムにおけるドライバ・ヒエラルキを、前記複数のドライバの他のものに対して変更する方法であって、
前記ドライバの中の特定の1つをリネームされるように指定するステップと、
前記リネームされたドライバとそれに関連するドライバとにサービスを固定するステップと、
前記ドライバの中の前記特定の1つをリネームしてリネームされたドライバを構成するステップと、
親関係にある前記関連するドライバを前記リネームされたドライバに更新し、前記リネームされたドライバを子ドライバとして反映し、子関係にある前記関連するドライバを前記リネームされたドライバに更新し、前記リネームされたドライバを親ドライバとして反映するステップと、
前記リネームされたドライバとその前記親類とへのサービスの固定を解除するステップと、
を含むことを特徴とする方法。

【請求項3】 個別の及び相互に関連する属性を有する複数のドライバを含むコンピュータ・システムにおけるドライバ・ヒエラルキを、前記複数のドライバの他のものに対して変更する方法であって、
前記ドライバの中の少なくとも交換されていない第1及び第2のものを指定して、前記個別の属性を交換するステップと、
前記交換されていない第1及び第2のドライバとそれに関連するドライバとにサービスを固定するステップと、
前記交換されていない第1及び第2のドライバの前記個別の属性を交換し、前記交換された第1及び第2のドライバの前記個別の属性をそれぞれ構成するステップと、
前記交換されていない第1のドライバに対して親関係にある前記関連するドライバを更新して前記交換された第

2のドライバを子のドライバとして反映し、前記交換されていない第2のドライバに対して子関係にある前記関連するドライバを更新して前記交換された第1のドライバを親のドライバとして反映するステップと、
を含むことを特徴とする方法。

【請求項4】 個別の及び相互に関連する属性を有する複数のドライバを含むコンピュータ・システムにおけるドライバ・ヒエラルキを、前記複数のドライバの他のものに対して変更する方法であって、
10 前記複数のドライバの中の少なくとも指定された1つに対して、利用可能なサービスのリストをコンパイルするステップと、
前記複数のドライバの中の前記指定された1つの前記個別の属性と、その親及び子のドライバの前記相互に関連する属性とを決定するステップと、
前記複数のドライバの中の前記指定された1つとそのドライバの前記親及び子のドライバとに対するサービスを固定するステップと、
前記複数のドライバの中の前記指定された1つとそのドライバの前記親及び子のドライバとの役割をスワップするステップと、
20 前記複数のドライバの中の前記指定された1つとそのドライバの前記親及び子のドライバとに対するサービスの固定を解除するステップと、
を含むことを特徴とする方法。

【請求項5】 コンピュータ可読のコードをその中に有するコンピュータ使用可能な媒体を含み、個別の及び相互に関連する属性を有する複数のドライバを含むコンピュータ・システムにおけるドライバ・ヒエラルキを、前記複数のドライバの他のものに対して変更するコンピュータ・プログラム製品であって、
コンピュータに、前記ドライバの中の特定の1つにその個別の属性を変更するように命令させるように構成されたコンピュータ可読のプログラム・コード・デバイスと、
コンピュータに、前記ドライバの中の特定の1つと前記ドライバの中の特定の1つに関連するものへのサービスを固定するように命令させるように構成されたコンピュータ可読のプログラム・コード・デバイスと、
40 コンピュータに、前記ドライバの中の前記特定の1つに前記個別の属性と、前記ドライバの中の前記特定の1つに前記関連するもののそれぞれの相互に関連する属性とを交換させるように構成されたコンピュータ可読のプログラム・コード・デバイスと、
コンピュータに、前記ドライバの中の前記特定の1つと前記ドライバの中の前記特定の1つに関連するものに対するサービスの固定を解除させるように構成されたコンピュータ可読のプログラム・コード・デバイスと、
50 を備えることを特徴とするコンピュータ・プログラム製品。

【請求項6】 コンピュータ可読のコードをその中に有するコンピュータ使用可能な媒体を含み、個別の及び相互に関連する属性を有する複数のドライバを含むコンピュータ・システムにおけるドライバ・ヒエラルキを、前記複数のドライバの他のものに対して変更するコンピュータ・プログラム製品であって、

コンピュータに、前記ドライバの中の特定の1つを指定するさせるように構成されたコンピュータ可読のプログラム・コード・デバイスと、

コンピュータに、前記リネームされたドライバとそれに関連するドライバとに対するサービスを固定させるように構成されたコンピュータ可読のプログラム・コード・デバイスと、

コンピュータに、前記ドライバの中の前記特定の1つをリネームしてリネームされたドライバを構成させるように構成されたコンピュータ可読のプログラム・コード・デバイスと、

コンピュータに、前記リネームされたドライバに対して親関係にある前記関連するドライバを更新して子ドライバとして前記リネームされたドライバを反映させ、前記リネームされたドライバに対して子関係にある前記関連するドライバを更新して親ドライバとして前記リネームされたドライバを反映させるように構成されたコンピュータ可読のプログラム・コード・デバイスと、

コンピュータに、前記リネームされたドライバと前記それと関連するドライバとに対する固定を解除させるように構成されたコンピュータ可読のプログラム・コード・デバイスと、

を備えることを特徴とするコンピュータ・プログラム製品。

【請求項7】 コンピュータ可読のコードをその中に有するコンピュータ使用可能な媒体を含み、個別の及び相互に関連する属性を有する複数のドライバを含むコンピュータ・システムにおけるドライバ・ヒエラルキを、前記複数のドライバの他のものに対して変更するコンピュータ・プログラム製品であって、

コンピュータに、前記ドライバの中の少なくとも交換されていない第1及び第2のものを指定して前記個別の属性を交換するように構成されたコンピュータ可読のプログラム・コード・デバイスと、

コンピュータに、前記交換されていない第1及び第2のドライバと前記それに関連するドライバとに対してサービスを固定させるように構成されたコンピュータ可読のプログラム・コード・デバイスと、

コンピュータに、前記交換されていない第1及び第2のドライバの前記個別の属性を交換して前記交換された第2及び第1のドライバそれぞれの前記個別の属性を構成させるように構成されたコンピュータ可読のプログラム・コード・デバイスと、

コンピュータに、前記交換されていない第1のドライバ

に対して親関係にある前記関連するドライバを更新して前記交換された第2のドライバを子ドライバとして反映させ、前記交換されていない第2のドライバに対して子関係にある前記関連するドライバを更新して前記交換された第1のドライバを親ドライバとして反映させるように構成されたコンピュータ可読のプログラム・コード・デバイスと、

コンピュータに、前記第1及び第2の交換されたドライバと前記それに関連するドライバとに対するサービスの固定を解除させるように構成されたコンピュータ可読のプログラム・コード・デバイスと、

を備えることを特徴とするコンピュータ・プログラム製品。

【請求項8】 コンピュータ可読のコードをその中に有するコンピュータ使用可能な媒体を含み、個別の及び相互に関連する属性を有する複数のドライバを含むコンピュータ・システムにおけるドライバ・ヒエラルキを、前記複数のドライバの他のものに対して変更するコンピュータ・プログラム製品であって、

コンピュータに、前記複数のドライバの中の少なくとも指定された1つに対して、利用可能なサービスのリストをコンパイルさせるように構成されたコンピュータ可読のプログラム・コード・デバイスと、

コンピュータに、前記複数のドライバの中の前記指定された1つの前記個別の属性と、その親及び子のドライバの前記相互に関連する属性とを決定させるように構成されたコンピュータ可読のプログラム・コード・デバイスと、

コンピュータに、前記複数のドライバの中の前記指定された1つとそのドライバの前記親及び子のドライバとに対するサービスを固定させるように構成されたコンピュータ可読のプログラム・コード・デバイスと、

コンピュータに、前記複数のドライバの中の前記指定された1つとそのドライバの前記親及び子のドライバとの役割をスワップさせるように構成されたコンピュータ可読のプログラム・コード・デバイスと、

コンピュータに、前記複数のドライバの中の前記指定された1つとそのドライバの前記親及び子のドライバとに対するサービスの固定を解除させるように構成されたコンピュータ可読のプログラム・コード・デバイスと、

を備えることを特徴とするコンピュータ・プログラム製品。

【請求項9】 個別の及び相互に関連する属性を有する複数のドライバを含むコンピュータ・システムにおけるドライバ・ヒエラルキを、前記複数のドライバの他のものに対して変更するシステムであって、

前記コンピュータ・システムのコンピュータ・オペレーティング・システムを前記複数のドライバに結合し、前記ドライバの中の特定のものにその個別の属性を変更させるメタディスク・ドライバと、

前記メタディスク・ドライバに応答し、前記ドライバの中の前記特定のもの前記ドライバの中の前記特定のものに関連するものに対するサービスを固定する第1の状態と、前記ドライバの中の前記特定のもの前記ドライバの中の前記特定のものに関連するものに対するサービスの固定を解除する第2の状態と、を有する固定手段と、

を備えており、前記メタディスク・ドライバは、前記固定手段を、前記ドライバの中の前記特定のもの前記個別の特性と前記ドライバの前記特性のものに関連するもののそれぞれの相互に関連する属性とが変更される間に前記第1の状態に導き、その後で、前記固定手段と前記第2の状態に導くことを特徴とするシステム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、この出願と同じ日付に米国で出願された「記憶装置マネージャと独立のコンフィギュレーション・インターフェース・トランスレータ及び方法」と題する米国特許出願第08/632217号の内容に関連する。この関連出願は、本出願の出願人であるサン・マイクロシステムズ社に譲渡されている。この出願の内容は、本出願において援用することにする。

【0002】本発明は、広くは、メタデバイス又はメタディスク・ドライバを介してコンピュータ・オペレーティング・システムに結合された複数のデバイス・ドライバを動作（オペレーティブ）に制御するコンピュータ及びコンピュータ・オペレーティング・システムの分野に関する。更に詳しくは、本発明は、コンピュータの大容量記憶装置のメタドライバと階層化された（layered）ドライバとのアイデンティティ（ID）をリネーム又は交換するシステム、方法、及びコンピュータ・プログラム製品に関する。この製品によれば、どの特定のドライバもそれ以外のドライバの私的（プライベート、個別）なデータ構造の知識を有する必要なく、分散的で協調的な態様で、ドライバ階層（ヒエラルキ）への変化のオンラインによる分散（分配）が可能になる。

【0003】

【従来の技術及び発明が解決しようとする課題】コンピュータ・システムのための現時点で利用可能なコンピュータの大容量記憶装置（マス・ストレージ）サブシステムは、一般的に、その全体的な性能、容量、管理の容易性、データ記憶装置の利用可能性などの点で、限定されている。更に、システム自体に関しては大容量記憶装置の動的な再構成がオンラインで可能であるにもかかわらず、このサブシステムでは、メタドライバすなわち下位の階層にあるドライバを、リネームしたり交換したりすることができない。

【0004】本発明は、機能的にはコンピュータ・オペレーティング・システムと1つ又は複数のメタドライバ

及び下の階層にあるドライバとの中間にあり、他のドライバのいずれかに関する私的なデータ構造の知識を有するなどの特定のドライバにも依存しないドライバのリネーム／交換機能を提供するメタディスク・ドライバ（又はディスク状のインターフェースを有するメタ記憶装置）を提供する。

【0005】

【課題を解決するための手段】すべてのメタデバイスのコンフィギュレーション（構成）及びステート（状態）に関する情報は、データの一体性を保証するために、複製された状態データベースの中に保存される。

【0006】本発明によって実現されるリネーム／交換技術は、マルチステージのトランザクションとして実行され、任意の幅又は深さを有するドライバのファミリー・ツリーに対して用いられるのに十分な程度の一般性を有する。重要なことは、このリネーム及び交換の機能は、下位の階層のデバイスがオンラインである間に行うことができ、また、動作が完了する又は任意の割り込みの発生において行われる限りは回復可能である原子的（atomic）な動作から構成されることである。

【0007】ここで提供されるシステム及び方法によって、任意のファイル・システム、データベース又はそれ以外のユーザ・データ、物理的なコンピュータ大容量記憶ディスク・ドライブのオンラインでの連結（concatenation）、ファイル・システムのオンラインでの拡張、ディスク・ストライピング（striping）、ホット・スペア（spares）、UNIX（米国登録商標）ファイル・システム（UFS）のログギング（logging）、及び安価なディスクの冗長アレー（RAID）サポートの最大で3通りのミラーリングが可能になる。

【0008】ここで開示する特定の実施例では、個別の及び相互に関連した属性（アトリビュート）を有する複数のドライバを含むコンピュータ・システムにおけるドライバ・ヒエラルキを、複数のドライバの他のものに対して変更する方法が提供される。この方法は、ドライバの中の特定の1つにその個別の属性を変更するように命じるステップと、サービスをその特定のドライバとそれと「親族関係にあるドライバ」（すなわち、それと「親」又は「子」の関係にあるドライバ）に固定（ロック）するステップと、を含む。また、この方法は、その特定のドライバの個別の属性と親類のそれぞれの相互に関連する属性とを変更するステップと、次に、その特定のドライバとその親類へのサービスの固定化を解除（アンロック）するステップと、を含む。

【0009】ここで開示する更に特定の実施例では、個別の及び相互に関連した属性を有する複数のドライバを含むコンピュータ・システムにおけるドライバ・ヒエラルキを、複数のドライバの中の他のものに対して変更する方法が提供される。この方法は、複数のドライバの中の少なくとも1つの指定されたドライバに対して利用可

能なサービスのリストをコンパイルするステップと、その指定されたドライバの別の属性と任意の親及び子のドライバの相互に関連する属性とを決定するステップとを有する。この方法は、更に、指定されたドライバとその親及び子のドライバに対してサービスを固定（ロック）するステップと、指定されたドライバとその親及び子のドライバとの役割を交換するステップと、指定されたドライバとその親及び子のドライバとに対するサービスの固定を解除するステップと、を含む。

【0010】

【発明の実施の態様】本発明が用いられる環境は、一般的な分散型計算システムに亘り、そこでは、汎用コンピュータ、ワークステーション、又はパーソナル・コンピュータが種々のタイプの通信リンクを介して、クライアント・サーバ構成に接続されており、多くはオブジェクトの形式であるプログラム及びデータは、このシステムの種々の構成メンバによって、このシステムの他の構成メンバによる実行及びアクセスに対して、利用可能になっている。汎用ワークステーション・コンピュータの構成要素のいくつかが図1に示されており、この図では、プロセッサ1は、入力／出力（I/O）部2と、中央処理装置（CPU）3と、メモリ部4とを有するように示されている。I/O部2は、キーボード5、ディスプレイ・ユニット6、ディスク記憶ユニット9、CD-ROMドライブ又はユニット7、又は、ワイド・エリア・ネットワーク（WAN）、ローカル・エリア・ネットワーク（LAN）、又はインターネットなどのそれ以外のネットワーク接続などのコンピュータ・ネットワーク11に接続される。CD-ROMユニット7は、典型的にはプログラム10及びデータを含むCD-ROM又はCD-ROM媒体8を読み出すことができる。本発明の装置及び方法を実行するメカニズムを含むコンピュータ・プログラム製品は、このシステムのメモリ部4、ディスク記憶ユニット9、CD-ROM8、又はネットワーク11に存在する。

【0011】図2を参照すると、本発明によるリネーム／交換技術を組み入れた代表的なコンピュータ・システム20が示されている。コンピュータ・システム20は、本発明の出願人であるサン・マイクロシステムズ社から市販されているSolaris（米国登録商標）オペレーティング・システムなどのUNIXベースのシステムの内容において、UNIXファイル・システム（UFS）を含むオペレーティング・システム22を含む。このようなUNIXベースのオペレーティング・システムは、また、DDI/DKIインターフェース26を含み、このインターフェースを通じて、ユーザ・アプリケーション28からのコールが、オペレーティング・システム22のコール及びサービスを介してなされる。

【0012】メタディスク・ドライバ30は、メタディスク・ドライバ30自身の対応するDDI/DKIイン

ターフェース26を介して、オペレーティング・システム22に結合される。オペレーティング・システム22とメタディスク・ドライバ30との間で移動される代表的なコールは、読み出し（リード）32、書き込み（ライト）34、及び入力／出力制御（IOCTL）36を含む。図解されていないが、オペレーティング・システム22とメタディスク・ドライバ30との間の他のコールも、従来型のUNIXベースのオペレーティング・システムに従えば、可能である。

10 【0013】図解されているように、メタディスク・ドライバ30には、また、関連するデータベース40に加えて、私的（プライベート）なサービス38に専用の部分も含まれ、これについては、以下でより詳細に説明する。メタディスク・ドライバ30は、メタディスク・ドライバ・インターフェースをより低いレベルのメタドライバ（又は、サブドライバ）に、例えば、IOCTL42、書き込み（ライト）44、及び読み出し（リード）46などのコールを介して、与える。そして、対応するメタディスク・ドライバは、また、図示されているように、対応するメタディスク・ドライバを含む。

20 【0014】メタディスク・ドライバ30は、ロード可能な疑似デバイス・ドライバの組として実現され、他の物理的デバイス・ドライバを用いて、入力／出力（I/O）リクエストをより下層のデバイスとやりとりする。メタディスク・ドライバ30は、ファイル・システム・インターフェースとデバイス・ドライバ・インターフェース26との間に存在し、上下の両方からの情報を解釈する。メタディスク・ドライバ30を通過した後で、情報は、ファイル・システムとデバイス・ドライバとの両方によって、予測される形式で受け取られる。メタディスク・ドライバ30は、ロード可能なデバイス・ドライバであり、他のどのデバイス・ドライバとも同じインターフェースを有している。メタディスク・ドライバ30の主要な要素には、メタデバイス、連結（concatenation）及びストライピング（striping）、ミラーリング（メタミラー及びサブミラーを含む）、UFSロギング（logging）（図示せず）、ホット・スペア（hot spare）（図示せず）、ディスクセット、及びRAIDデバイスが含まれる。

40 【0015】メタデバイスは、メタディスク・ドライバ30の基本的な機能ユニットであり、物理的なディスク・パーティション又は「スライス」のように用いることができる。これらの論理デバイスは、単一のデバイス、ストライプの連結、又はデバイスのストライプを用いるように構成できる1つ又は複数の成分パーティションから作ることができる。連結又はストライプを含むメタデバイスは記憶装置の容量を増加させ、他方で、ミラーリング及びUFSロギングは可用性（アベイラビリティ）を向上させる。ストライピングは、性能（パフォーマンス）を向上させる。いずれの場合にも、メタデバイス

は、ユーザ・アプリケーション28ソフトウェアと、構成要素コントローラ・ハードウェアとに対して、透過的（トランスペアレント）である。これらは、すべてのSPARC（米国登録商標）システム上のIPI及びSCSIデバイスから、そして、すべてのx86システム上のSCSI及びIDEデバイス上に構成することができる。

【0016】それぞれのメタデバイスは、連結又は成分（コンポーネント）パーティションのストライプのどちらかである。連結とストライプとは、共同して作用し、2つ又はそれより多くのスライスを連結し、2つのより大きなスライスを作成する。パーティションが連結されるときには、成分ブロックのアドレッシングは、シーケンシャルに行われる。ファイル・システムなどのオペレーティング・システム22のサービスは、全体の連結を用いることができる。

【0017】データベース40は、ステート・データベースのレプリカを維持し、すべてのメタドライブ、メタデバイス、メタミラー、メタトランス・デバイス、ホット・スペア、及びRAIDデバイスに対するコンフィギュレーション及びステータス情報を追跡するのに必要な不揮発性の記憶装置を提供する。レプリカは、また、生じたエラー条件も追跡する。メタデバイスが構成された後では、メタデバイス・ドライブは、そのコンフィギュレーション及びステータス情報を記憶していることが必要である。メタデバイスのステート・データベース40は、メタデバイス・ドライブの長期メモリである。メタデバイス・ドライブは、すべてのメタデバイス・コンフィギュレーション情報を、ステート・データベース40に記憶するが、これには、ミラー、サブミラー、連結、ストライプ、メタトランス・デバイス、及びホット・スペアに関するコンフィギュレーション情報が含まれる。

【0018】複製されたメタデバイス・ドライブのステート・データベース40が失われる場合には、このメタデバイス・ドライブは、どのようなコンフィギュレーション情報を知るための方法を有さず、結果的に、メタデバイス上に記憶されたすべてのデータの損失が生じる。ハードウェアの故障が原因であるメタデバイスのステート・データベースの損失に対する保護のために、ステート・データベース40の複数のレプリカ（又は、コピー）が維持される。3つのレプリカによって、システムのクラッシュの結果として生じ得る破壊に対して、ステート・データベース40が保護される。ステート・データベース40のそれぞれのレプリカは、チェックサムを含む。ステート・データベース40が更新されるときには、それぞれのレプリカは、一度にひとつが、修正される。データベース40が更新されている間にクラッシュが生じる場合には、レプリカの中の1つだけが破壊される。システムがリブートするときには、メタデバイス・ドライブは、レプリカの中に埋め込まれたチェックサム

を用いて、レプリカが破壊されたかどうかを判断する。破壊されたレプリカは、すべて、無視される。

【0019】メタデバイスのステート・データベース40を含むディスクがオフになる場合には、メタデバイスは完全に機能する状態に保たれるが、その理由は、データベースが依然として動作しているレプリカの中の1つから検索されるからである。リブート動作の後のコンフィギュレーションへの変更は、システムが回復するときには動作しているレプリカにだけ記憶される。オフに切り換えられたディスク・ドライブが後に再びオンになる場合には、そのディスク上に記憶されたレプリカに含まれるデータは、他のレプリカとの比較の後で無視する。

【0020】メタディスク・ドライブ30と共に用いることのできるメタデバイス・ドライブすなわちメタドライブの例は、ストライプ・メタドライブ48、ミラー・メタドライブ50、RAIDメタドライブ52、及びトランス・メタドライブ54である。

【0021】ストライプ・メタドライブ48は、DDI/DKIインターフェースを、例えば、スライス（又は、パーティション）階層化されたドライブ56に与える。他方で、ミラー・メタドライブ50は、1対の下位のストライプ階層化されたドライブ58、60に結合され得る。ストライプ階層化されたドライブ58、60は、DDI/DKIインターフェースを介して、対応するスライス階層化されたドライブ62、64に結合され得る。

【0022】ストライピング（ストライプ化）は連結（コンカテネーション）と類似しているが、メタデバイス・ブロックのアドレッシングが、シーケンシャルになされるのではなく成分に対してインターレースされる点が異なる。ストライプが定義されるときには、インターレースのサイズが特定され、それによって、ストライプの次の成分に移動する前にどれだけ多くのデータが成分の上に配置されるかを判断する。

【0023】データはストライプに亘って広がっているため、読み出し及び書き込みが複数のディスク・アームに亘って広がっているという事実のために、パフォーマンスの向上が認識される。更に、同時的なI/Oリクエストは、連結の場合と同様に、別のディスク・アームを用いることができる。

【0024】ミラーリングもまた、例えば、ミラー・メタドライブ50などの別個のメタデバイス上でサポートされる。これによって、システムは、2通りのミラーリングを用いて単一の構成要素の故障に対して、及び、3通りのミラーリングを用いて二重の故障に対して、フォールト・トレラントであることが可能になる。ミラーリングは、また、ファイル・システムのオンラインでのバックアップに対しても用いることができる。

【0025】ミラーリングは、1つ又は複数の他のメタデバイスから構成される特別のタイプのメタデバイスで

あるメタミラーを作成することによって、設定される。メタミラーの内部のそれぞれのメタデバイスは、サブミラーと称される。メタミラーがいったん定義されると、追加的なサブミラーを、システムをダウンさせずに、また、既存のメタミラーに対する読み出し及び書き込みを中断せずに、追加することができる。サブミラーが付属すると、そのメタミラーの中の別のサブミラーからのデータは、すべてが、リシンチング (resynching) と称するプロセスにおいて、新たに付属したサブミラーに、自動的に書き込まれる。いったんリシンチングが完了すると、新たなサブミラーは、読み出し及び書き込み可能である。

【0026】RAIDメタドライブ52は、多数の下位のスライス階層化されたドライブ66、68、70への結合のためのDDI/DKIインターフェースを与える。この代表的な例ではスライス階層化されたドライブ66、68、70を3つだけ含むように図解されているが、このような階層化されたドライブは、所望のRAIDセット及びRAIDレベルに応じて、任意の数を含むようにしてよい。

【0027】RAIDデバイスは、図解されているスライス66-70のような、3つ以上の物理的なパーティションすなわちスライスを含み得る。それぞれのパーティションは、一般的に、コラムと称される。RAIDメタドライブ52は、追加的なパーティションをそのメタデバイスに連結することによって、成長させることが可能である。RAID5は、複数の物理的パーティションを用いて、単一の大きなスライスをシミュレートする。1つ又は複数の物理的スライス上の単一のセクタは、1つのセクタ分の隣接するデータか、そのアレーにおけるすべての他のスライスの同じセクタ上のデータに関連するパリティ情報かのどちらかを含む。RAID5のコンフィギュレーションは、単一のディスク故障から回復することができ、一般的に、ディスクのミラーリングよりもコスト的に効率的である。

【0028】トランス・メタドライブ54は、1対のストライプ階層化されたドライブ74、76とスライス階層化されたドライブ78、80と順に有する下位のミラー階層化されたドライブ72に結合することができる。トランス・メタドライブ54はまた、1つ又は複数の下位のスライス階層化されたドライブ84に加えて、下位のRAID階層化されたドライブ82を有する。トランス・メタドライブ54はまた、示されているように、下位のスライス階層化されたドライブ86に直接に結合することができる。

【0029】次に、更に、図3を参照すると、先の図面に図解されている多数のメタドライブ及び/又は階層化されたドライブの簡略化された概念的な表現が示されている。図3の図解は、ドライブと(先行する図面においてより詳細に示されるように)結果的にドライブ・ヒエ

ラルキの変更を生じるその後のリネーム及び交換動作との間の「自分自身」、「親」、及び「子」の関係を図解するのに有用である。この点で、メタデバイス100は、D100の自己指定を有し、他方で、メタデバイス100の子メタデバイス102は、D10の指定を有する。次に、メタデバイス102の更なる子メタデバイス104は、D1の指定を有する。同時に、メタデバイス102はメタデバイス104の親であり、メタデバイス100はメタデバイス102の親である。

【0030】同様に、他の独立のメタデバイス間の関係が示されており、D20の自己指定を有するメタデバイス106は、指定D2を有するメタデバイス108に対して親関係にある。つまり、メタデバイス108は、メタデバイス106の子である。D3の自己指定を有する単一の独立のメタデバイス110は、付随する親又は子関係にあるメタデバイスを有していないものとして示されている。

【0031】次に、図4のAを参照すると、図3に示されたメタデバイス及び階層化されたドライブの簡略化された概念図が、リネーム動作において用いられ、メタデバイス100の自己識別がD100からD99に変更されている。本願で開示されているリネーム/交換技術を用いて、メタデバイス100は次にその自己指定をD99に更新し、その子メタデバイス102(D10と指定されている)は、次に、その親を、メタデバイス100からD99へのリネームを反映するように更新する。

【0032】次に、図4のBを参照すると、図3に示されたドライブ・ヒエラルキの当初のコンフィギュレーションに基づいて、例示的なリネーム動作が、更に示されている。この代表的な例では、メタデバイス102は、D10からD9にリネームされており、指定の変更を反映するために自己のリネームが必要になっている。しかし、この例では、メタデバイス100(D100と指定)は、メタデバイス102に対して親関係にあり、従って、その複数の子を更新して、残りのメタデバイス102のD9へのリネームを反映する。同様に、メタデバイス104は、メタデバイス102に対して子関係にあり、その親の指定を更新して、メタデバイスのD9への名称の変更を反映する。

【0033】次に図5のAを参照すると、本発明による代表的な交換動作が示されており、図3のメタデバイス106(前はD20)とメタデバイス108(前はD2)は、メタデバイス106はD2(親)に、メタデバイス108はD20(子)になるように、自己識別を交換する。換言すると、メタデバイス106(図3)は、そのソースを下方向に更新し、それに対して、メタデバイス108(図3)は、その目標(デスティネーション)を上方向に更新する。

【0034】次に、図5のBを参照すると、交換動作の別の例示的な図解が示されており、メタデバイス10

2、104は、メタデバイス102はアイデンティティD1を保持し、メタデバイス104はメタデバイス102の先のアイデンティティD10を保持するように、自己アイデンティティを交換する。換言すれば、メタデバイス104は、その目標を上方向に更新し、それに対して、メタデバイス102は、そのソースを下方向に更新する。この図解では、メタデバイス112(D5)はメタデバイス102の子であり続けるが、その親の指定は、D10ではなくD1である。

【0035】図6を参照すると、本発明によるシステム及び方法の役割スワップ・サービスが、リネーム(円で囲まれたR)及び交換(円で囲まれたE)動作の両方に関して図解されており、メタドライバ及び/又は階層化されたドライバ・ヒエラルキを更新するのに用いられている。図6の表の形式での表現は、次の表1に表されたプロセスの部分6を形成している。

【0036】

【表1】

表1

MD リネーム/交換

名称付きサービス

- 1-親をリストする
- 2-自分自身をリストする
- 3-子をリストする
- 4-ロック(固定)する
- 5-チェックする

6-更新

親

子

自分自身を上方向に

自分自身を下方向に

ソースを下方向に

ソースを上方向に

デスティネーションを上方向に

ソースを下方向に

デスティネーションを下方向に

7-固定を解除

【0037】従って、本発明は、機能的にはコンピュータのオペレーティング・システムと1つ又は複数のメタドライバ及び下の階層のドライバとの中間にあり、他のドライバのどれかの私的なデータ構造の知識を有するなどの特定のドライバにも依存しないドライバのリネーム/交換機能を与えるメタディスク・ドライバ(又は、ディスク状のインターフェースを有するメタ記憶装置)を与える。すべてのメタデバイスのコンフィギュレーション及びステートに関する情報は、複製されたステート・デ

ータベースに保存され、データの一体性が保証される。

【0038】以上で本発明の原理を特定のオペレーティング・システムとドライバとに関して説明したが、以上の説明は単に例示目的のものであって、明らかに、発明の技術的範囲を制限するものではない。特に、以上の説明から得られる発明に対して、当業者であれば、修正を行うことができよう。そのような修正は、それ自体が既に既知でありここで開示した特徴に加えて用いることができる他の特徴を含み得る。特許請求の範囲は、この出願においては、特徴の特定の組合せとして記載されているが、本発明の範囲は、本出願において明示的に又は暗示的に開示された個々の特徴の組合せである任意の新たな特徴を含む可能性があり、その一般化又は修正は、当業者には明らかである。これは、そのような一般化又は修正がいずれかの請求項に関係するか、または、本発明が直面する技術的な課題から導かれるかに限らない。従って、出願人は、出願手続きにおいて、そのような特徴に関する新たな請求項を追加する、又は、分割出願を行う権利を留保しているものとする。

【図面の簡単な説明】

【図1】本発明の動作環境の一部分を形成する汎用コンピュータの簡略化された図解である。

【図2】複数のメタドライバと下の階層にあるドライバとにメタディスク・ドライバを介して動作的に結合し本発明のリネーム/交換技術を実現するデバイス・ドライバ・インターフェースを組み入れた例示的なコンピュータ・プログラムのオペレーティング・システムの簡略化された概念図である。

【図3】図2に示した多数のメタドライバ及び階層化されたドライバの簡略化された概念図であり、結果的に先行する図面に示されたようなドライバ・ヒエラルキへの変化を生じる後続のリネーム及び交換動作におけるドライバの間の「自分自身」、「親」、及び「子」の関係を図解するのに有用である。

【図4】図4のA及びBは、共に、ドライバの中のあるもののリネームの結果として生じるメタドライバ及び階層化されたドライバ・ヒエラルキへの可能な変化を図解しており、ここでは、「自分自身」及び「子」への変化と、「自分自身」、「親」、及び「子」の関係とがそれぞれ生じる。

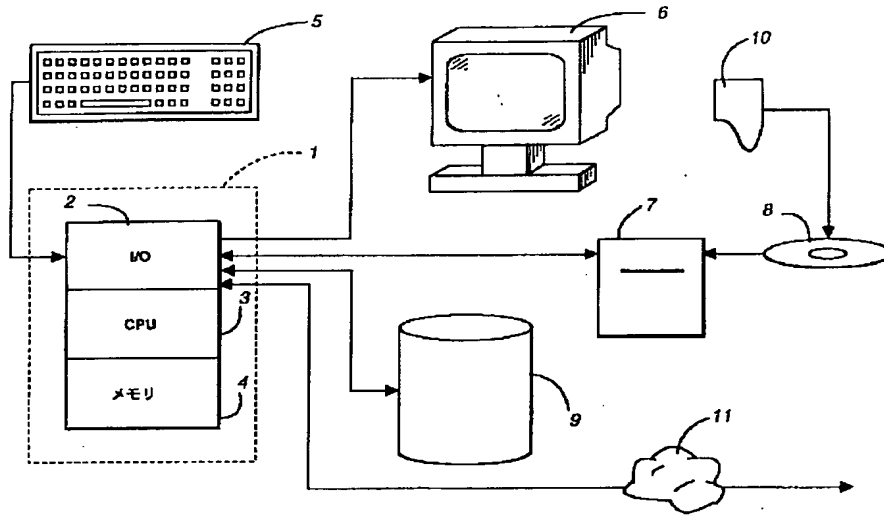
【図5】図5のA及びBは、共に、2つのドライバの間でのドライバの中のあるドライバの交換の結果として生じるメタドライバ及び階層化されたドライバ・ヒエラルキへの他の可能な変化を図解しており、ここでは、「自分自身」、「親」、及び「子」の関係が変更され、また、他の2つのドライバの間の場合であって、ここでは、交換は、別のドライバの「親」に関する。

【図6】リネーム及び交換動作の両方に関してここで開示されている本発明のシステム及び方法の役割交換サービスによって実現されるメタドライバ及び階層化された

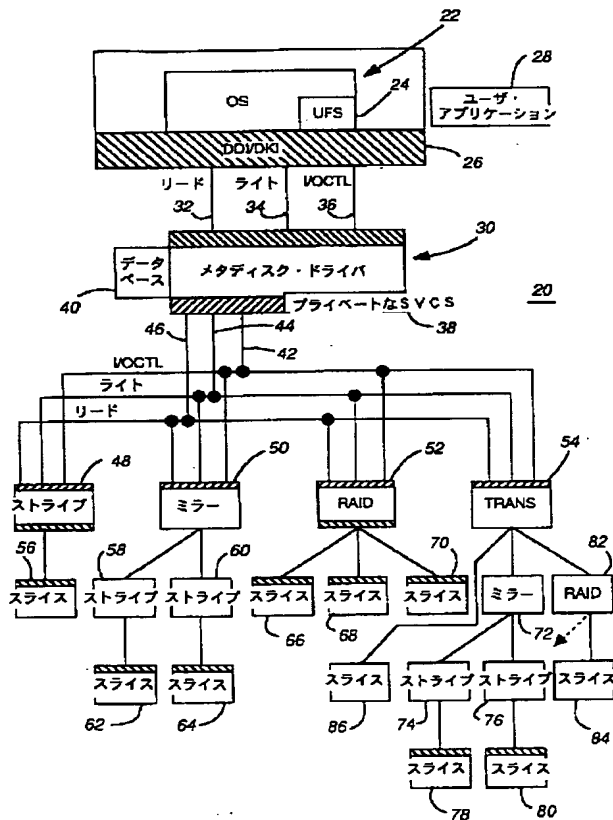
ドライバ・ヒエラルキへの更新を、表にして表したもの

である。

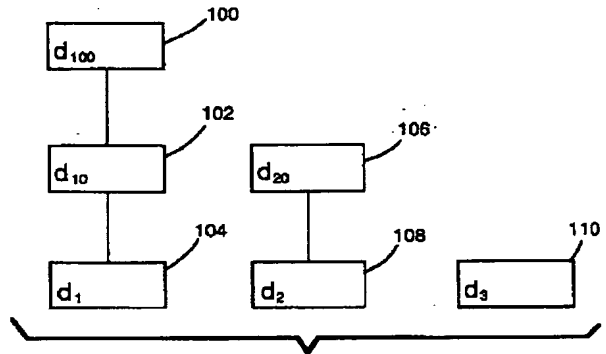
【図1】



【図2】



【図3】



【図4】

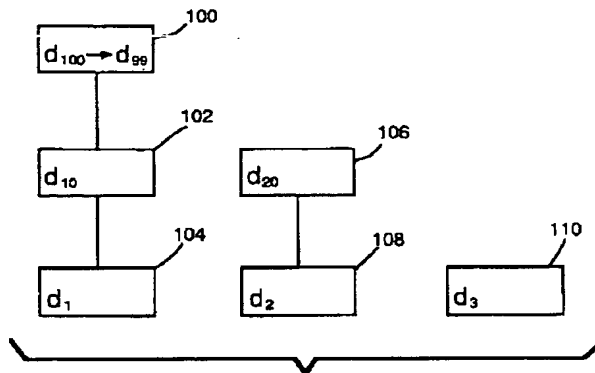


Fig. 4A

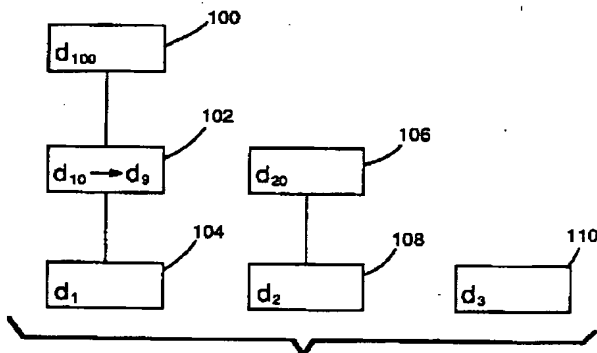


Fig. 4B

【図5】

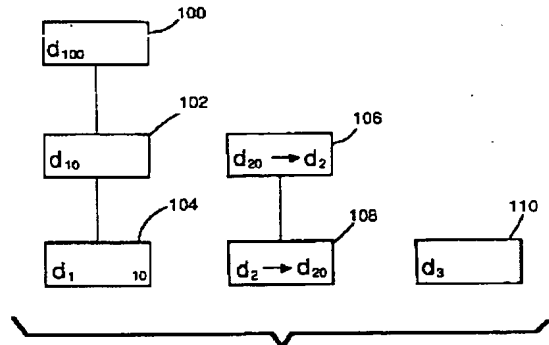


Fig. 5A

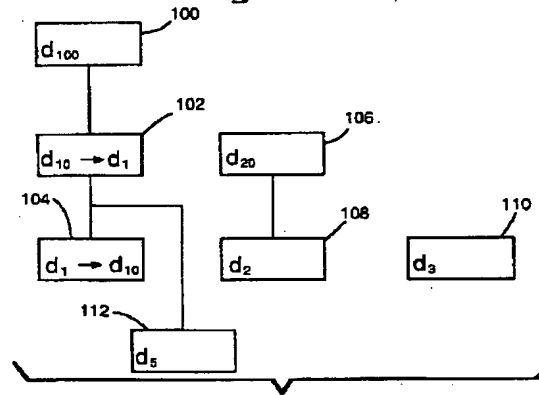


Fig. 5B

【図6】

		前の役割		
		親	自分自身	子
新たな役割	親	子を更新 (E) (R)	ソースを 下方向に更新 (E)	—
	自分自身	デスティネーションを 上方向に更新 (E)	自分自身を更新 (R)	デスティネーションを 上方向に更新 (E)
	子	—	ソースを 下方向に更新 (E)	親を更新 (E) (R)

フロントページの続き

(71)出願人 597004720

2550 Garcia Avenue, MS
PAL1-521, Mountain V
iew, California 94043-
1100, United States of
America

(72)発明者 スティーブン・ティー・セネター
アメリカ合衆国コロラド州80920, コロラ
ド・スプリングス, ウェストミンスター・
ドライブ 8625

(72)発明者 デール・アール・パスモア
アメリカ合衆国コロラド州80919, コロラ
ド・スプリングス, ストニー・ポイント・
コート 1625

(72)発明者 ロバート・エス・ギッティンズ
アメリカ合衆国コロラド州80866, ウッド
ランド・パーク, キングズ・クラウン・ロ
ード 1020